UNITED STATES PATENT APPLICATION

FOR

SIGNAL FORMAT THAT FACILITATES EASY SCALABILITY OF ENCRYPTED STREAMS

Inventors:
SUSIE J. WEE
JOHN G. APOSTOLOPOULOS

Prepared by:
WAGNER, MURABITO & HAO LLP
Two North Market Street
Third Floor
San Jose, California 95113

SIGNAL FORMAT THAT FACILITATES EASY SCALABILITY OF ENCRYPTED STREAMS

TECHNICAL FIELD

The present claimed invention relates to the field of streaming media. More specifically, the present claimed invention relates to the encoding and decoding of data.

BACKGROUND ART

Wireless streaming environments present many challenges for the system designer. For instance, clients can have different display, power, communication, and computational capabilities. In addition, wireless communication links can have different maximum bandwidths, quality levels, and time-varying characteristics. A successful wireless video streaming system must be able to stream video to heterogeneous clients over time-varying wireless communication links, and this streaming must be performed in a scalable and secure manner. Scalability is needed to enable streaming to a multitude of clients with different device capabilities. Security is particularly important in wireless networks to protect content from eavesdroppers.

In order to achieve scalability and efficiency in wireless streaming environments, one must be able to easily adapt or transcode the compressed video stream at intermediate network nodes. A transcoder takes a compressed video system as the input, then processes it to produce another compressed video stream as the output. Sample transcoding operations include bitrate reduction, rate shaping, spatial downsampling, frame rate reduction, and changing compression formats. Network transcoding can improve system scalability and efficiency, for example, by adapting the spatial resolution of a video stream for a particular client's display capabilities or by dynamically adjusting the bitrate of a video stream to match a wireless channel's time-varying characteristics.

While network transcoding facilitates scalability in video streaming systems, it also presents a number of challenges. First, while

computationally efficient transcoding algorithms have been developed, even these are not well-suited for processing hundreds or thousands of streams at intermediate wired network nodes or even a few streams at intermediate low-power wireless networking relay nodes. Furthermore, network transcoding poses a serious threat to the security of the streaming system because conventional transcoding operations performed on encrypted streams generally require decrypting the stream, transcoding the decrypted stream, and then re-encrypting the result. Because every transcoder must decrypt the stream, each network transcoding node presents a possible breach in the security of the entire system.

More specifically, in conventional video streaming approaches employing application-level encryption, video is first encoded into a bitstream using interframe compression algorithms. These algorithms include, for example, the Moving Picture Experts Group (MPEG) standard, the International Telecommunications Union (ITU) standard, H.263, or intraframe compression algorithms such as, for example, the Joint Photographic Experts Group (JPEG) or JPEG2000 standards. The resulting bitstream is then encrypted, and the resulting encrypted stream is packetized and transmitted over the network using a transport protocol such as unreliable datagram protocol (UDP). Prior Art Figure 1 is a block diagram 100 which illustrates the order in which conventional application-level encryption is performed (i.e. Encode 102, Encrypt 104 and Packetize 106). One difficulty with this conventional approach arises when a packet is lost. Specifically, error recovery is difficult because without the data from the lost packet, decryption and/or decoding may be difficult if not impossible.

Prior Art Figure 2 is a block diagram 200 illustrating another conventional secure video streaming system that uses network-level encryption (i.e. Encode 202, Packetize 204, and Encrypt 206). The system of Prior Art Figure 2 can use the same video compression algorithms as the system of Prior Art Figure 1. However, in the system of Prior Art Figure 2, the packetization can be performed in a manner that considers the content of the coded video and thus results in better error recovery, a concept known to the networking community as application-level framing. For example, a common approach is to use MPEG compression with the RTP transport protocol which is built on unreliable datagram

protocol (UDP), RTP provides streaming parameters such as time stamps and suggests methods for packetizing MPEG payload data to ease error recovery in the case of lost or delayed packets. However, error recovery is still difficult and without data from a lost packet, decryption and/or decoding is still difficult if not impossible.

Both of the conventional approaches of Prior Art Figure 1 and Prior Art Figure 2 are secure in that they transport the video data in encrypted form. However, with these conventional approaches, if network transcoding is needed, it must be performed in accordance with the method of Prior Art Figure 3. That is, as shown in block diagram 300, the necessary transcoding operation is a decrypt 302, decode 304, process 306, re-encode 308, and re-encrypt 310 process. As shown in the block diagram 400 of Prior Art Figure 4, in another conventional approach, the computational requirements of the operation of Prior Art Figure 3 are reduced to a decrypt 402, transcode 404, and re-encrypt 406 process. Specifically, this computational reduction is achieved by incorporating and efficient transcoding algorithm (i.e. transcode module 404) in place of the decode 304, process 306, and re-encode 308 modules of Prior Art Figure 3. However, even such improved conventional transcoding algorithms have computational requirements that are not well-suited for transcoding many streams in a network node. Furthermore, a more critical drawback stems from the basic need to decrypt the stream for every transcoding operation. As, mentioned above, each time the stream is decrypted, it opens another possible attack point and thus increases the vulnerability of the system. Thus, each transcoder further threatens the security of the overall system.

As yet another concern, wireless streaming systems are limited by wireless bandwidth and client resources. Wireless bandwidth is scarce because of its shared nature and the fundamental limitations of wireless spectrum. Client resources are often practically limited by power constraints and by display, communication, and computational capabilities. As an example, wireless transmission and even wireless reception alone typically consume large power budgets. In order to make the most efficient use of wireless bandwidth and client resources, it is desirable to send clients the lowest bandwidth video streams that match their display and communication capabilities. In wireless streaming systems where a sender streams video to a number of heterogeneous

clients with different resources, network transcoders can be used to help achieve end-to-end system efficiency and scalability.

In hybrid wired/wireless networks, it is often necessary to simultaneously stream video to fixed clients on a wired network and to mobile clients on a wireless network. In such a hybrid system, it may often be desirable to send a full-bandwidth, high-resolution video stream to the fixed wired client, and a lower-bandwidth, medium-resolution video stream to the mobile wireless receiver. Conventional video streaming approaches, however do not achieve the efficiency, security, and scalability necessary to readily accommodate the video streaming corresponding to hybrid wired/wireless networks.

Yet another example of the drawbacks associated with conventional video streaming approaches is demonstrated in conjunction with wireless appliance networks. In many wireless appliance networks, mobile senders and receivers communicate with one another over wireless links. A sender's coverage area is limited by the power of the transmitted signal. Relay devices can be used to extend the wireless coverage area when intended receivers are beyond the immediate coverage area of the sender. However, in the case of heterogeneous clients within the same wireless network, it may be desired to provide a higher bandwidth, high-resolution video stream to the high power wireless receivers, and a lower bandwidth, low-resolution video stream to the low power wireless receivers. Once again, conventional video streaming approaches, however do not achieve the efficiency, security, and scalability necessary to readily accommodate such video streaming demands in wireless appliance networks. Although the above-listed discussion specifically mentions the shortcomings of prior art approaches with respect to the streaming of video data, such shortcomings are not limited solely to the streaming of video data. Instead, the problems of the prior art span various types of media including, but not limited to, audio-based data, image-based data, web page-based data, and the like.

Thus, the need has arisen for a data packet format which enables secure and scalable encoding, transcoding, and decoding of data.

DISCLOSURE OF THE INVENTION

The present invention provides, in one embodiment, a data packet format which enables secure and scalable encoding, transcoding, and decoding of data.

Specifically, in one embodiment, the present invention is comprised of a computer readable medium having a data packet stored therein for causing a functional change in the operation of a device is disclosed. In one embodiment, the data packet is comprised of a scalably encoded, progressively encrypted data portion. In the present embodiment, the data packet further includes a header data portion corresponding to the scalably encoded, progressively encrypted data portion. The header data portion includes information adapted to be used by a transcoder to efficiently transcode the scalably encoded, progressively encrypted data portion.

In yet another embodiment, the data packet includes the same features as the above-described embodiment, but the data header portion of the data packet is encrypted.

These and other technical advantages of the present invention will no doubt become obvious to those of ordinary skill in the art after having read the following detailed description of the preferred embodiments which are illustrated in the various drawing figures.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and form a part of this specification, illustrate embodiments of the invention and, together with the description, serve to explain the principles of the invention:

PRIOR ART FIGURE 1 a block diagram which illustrates the order in which conventional application-level encryption is performed.

PRIOR ART FIGURE 2 is a block diagram which illustrates another conventional secure video streaming system using network-level encryption.

PRIOR ART FIGURE 3 is block diagram illustrating a conventional transcoding method.

PRIOR ART FIGURE 4 is block diagram illustrating another conventional transcoding method.

FIGURE 5 is a schematic diagram of an exemplary computer system used to perform steps of the present method in accordance with various embodiments of the present claimed invention.

FIGURE 6 is a flow chart of steps performed in an efficient, secure, and scalable encoding method in accordance with one embodiment of the present claimed invention.

FIGURE 7 is a block diagram of an encoding system in accordance with one embodiment of the present claimed invention.

FIGURE 8 is a block diagram of an encoding system having a video prediction unit (VPU) coupled thereto in accordance with one embodiment of the present claimed invention.

FIGURE 9 is a block diagram of an encoding system having a video prediction unit (VPU) integral therewith in accordance with one embodiment of the present claimed invention.

FIGURE 10A is a schematic depiction of a frame of video data in

accordance with one embodiment of the present claimed invention.

FIGURE 10B is a schematic depiction of the frame of video data of FIGURE 10A after segmentation into corresponding tiles in accordance with one embodiment of the present claimed invention.

FIGURE 10C is a schematic depiction of the frame of video data of FIGURE 10A after segmentation into corresponding non-rectangular regions in accordance with one embodiment of the present claimed invention.

FIGURE 10D is a schematic depiction of the frame of video data of FIGURE 10A after segmentation into corresponding overlapping non-rectangular regions in accordance with one embodiment of the present claimed invention.

FIGURE 11 is a flow chart of steps performed in decoding video data which has been efficiently, securely, and scalably, encoded in accordance with one embodiment of the present claimed invention.

FIGURE 12 is a block diagram of a decoding system in accordance with one embodiment of the present claimed invention.

FIGURE 13 is a block diagram of a decoding system having a video prediction unit (VPU) coupled thereto in accordance with one embodiment of the present claimed invention.

FIGURE 14 is a block diagram of a decoding system having a video prediction unit (VPU) integral therewith in accordance with one embodiment of the present claimed invention.

FIGURE 15A is a block diagram of an exemplary hybrid wired/wireless network upon which embodiments of the present invention may be practiced.

FIGURE 15B is a block diagram of an exemplary wireless network upon which embodiments of the present invention may be practiced.

FIGURE 16 is a block diagram of a source node, an intermediate

(transcoder) node, and a receiving node in accordance with one embodiment of the present invention.

FIGURE 17 is a block diagram of one embodiment of a transcoder device upon which embodiments of the present invention may be practiced in accordance with one embodiment of the present claimed invention.

FIGURES 18A, 18B, 18C, 18D and 18E are data flow diagrams illustrating various embodiments of a method for transcoding data packets in accordance with one embodiment of the present claimed invention.

FIGURE 19 is a flowchart of the steps in a process for transcoding data packets in accordance with one embodiment of the present claimed invention.

FIGURE 20 is a schematic representation of a data packet including header data and scalably encoded, progressively encrypted video data in accordance with one embodiment of the present claimed invention.

FIGURE 21 is a schematic representation of a data packet including scalably encoded, progressively encrypted video data in accordance with one embodiment of the present claimed invention.

The drawings referred to in this description should be understood as not being drawn to scale except if specifically noted.

BEST MODES FOR CARRYING OUT THE INVENTION

Reference will now be made in detail to the preferred embodiments of the invention, examples of which are illustrated in the accompanying drawings. While the invention will be described in conjunction with the preferred embodiments, it will be understood that they are not intended to limit the invention to these embodiments. On the contrary, the invention is intended to cover alternatives, modifications and equivalents, which may be included within the spirit and scope of the invention as defined by the appended claims. Furthermore, in the following detailed description of the present invention, numerous specific details are set forth in order to provide a thorough understanding of the present invention. However, it will be obvious to one of ordinary skill in the art that the present invention may be practiced without these specific details. In other instances, well known methods, procedures, components, and circuits have not been described in detail as not to unnecessarily obscure aspects of the present invention.

It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the following discussions, it is appreciated that throughout the present invention, discussions utilizing terms such as "receiving", "segmenting", "scalably encoding", "progressively encrypting" or the like, refer to the actions and processes of a computer system, or similar electronic computing device. The computer system or similar electronic computing device manipulates and transforms data represented as physical (electronic) quantities within the computer system's registers and memories into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission, or display devices. The present invention is also well suited to the use of other computer systems such as, for example, optical and mechanical computers.

COMPUTER SYSTEM ENVIRONMENT OF THE PRESENT SECURE SCALABLE STREAMING INVENTION

With reference now to Figure 5, portions of the present interrupt events chaining method and system are comprised of computer-readable and computer-executable instructions which reside, for example, in computer-usable media of a computer system. Figure 5 illustrates an

exemplary computer system 500 used in accordance with one embodiment of the present secure scalable streaming invention. It is appreciated that system 500 of Figure 5 is exemplary only and that the present invention can operate on or within a number of different computer systems including general purpose networked computer systems, embedded computer systems, routers, switches, server devices, client devices, various intermediate devices/nodes, stand alone computer systems, and the like. Additionally, computer system 500 of Figure 5 is well adapted having computer readable media such as, for example, a floppy disk, a compact disc, and the like coupled thereto. Such computer readable media is not shown coupled to computer system 500 in Figure 5 for purposes of clarity.

System 500 of Figure 5 includes an address/data bus 502 for communicating information, and a central processor unit 504 coupled to bus 502 for processing information and instructions. Central processor unit 504 may be an 80x86-family microprocessor. System 500 also incudes data storage features such as a computer usable volatile memory 506, e.g. random access memory (RAM), coupled to bus 502 for storing information and instructions for central processor unit 504, computer usable non-volatile memory 508, e.g. read only memory (ROM), coupled to bus 502 for storing static information and instructions for the central processor unit 504, and a data storage unit 510 (e.g., a magnetic or optical disk and disk drive) coupled to bus 502 for storing information and instructions. System 500 of the present invention also includes an optional alphanumeric input device 512 including alphanumeric and function keys coupled to bus 502 for communicating information and command selections to central processor unit 504. System 500 also optionally includes an optional cursor control device 514 coupled to bus 502 for communicating user input information and command selections to central processor unit 504. System 500 of the present embodiment also includes an optional display device 516 coupled to bus 502 for displaying information.

Referring still to Figure 5, optional display device 516 of Figure 5, may be a liquid crystal device, cathode ray tube, or other display device suitable for creating graphic images and alphanumeric characters recognizable to a user. Optional cursor control device 514 allows the computer user to dynamically signal the two dimensional movement of a visible symbol (cursor) on a display screen of display device 516. Many

implementations of cursor control device 514 are known in the art including a trackball, mouse, touch pad, joystick or special keys on alphanumeric input device 512 capable of signaling movement of a given direction or manner of displacement. Alternatively, it will be appreciated that a cursor can be directed and/or activated via input from alphanumeric input device 512 using special keys and key sequence commands. The present invention is also well suited to directing a cursor by other means such as, for example, voice commands. A more detailed discussion of the present secure scalable streaming invention is found below.

GENERAL DESCRIPTION OF THE PRESENT SECURE SCALABLE STREAMING INVENTION

With reference next to Figure 6, Figure 11, and Figure 19, flow charts 600, 1100, and 1900, respectively, illustrate exemplary steps used by the various embodiments of present invention. Flow charts 600, 1100, and 1900 includes processes of the present invention which, in one embodiment, are carried out by a processor under the control of computer-readable and computer-executable instructions. The computer-readable and computer-executable instructions reside, for example, in data storage features such as computer usable volatile memory 506, computer usable non-volatile memory 508, and/or data storage device 510 of Figure 5. The computer-readable and computer-executable instructions are used to control or operate in conjunction with, for example, central processing unit 504 of Figure 5.

As an overview, the present invention is directed towards any data which can be scalably encoded and, specifically, any data that combines scalable encoding with progressive encryption. For purposes of the present Application, scalable coding is defined as a process which takes original data as input and creates scalably coded data as output, where the scalably coded data has the property that portions of it can be used to reconstruct the original data with various quality levels. Specifically, the scalably coded data is often thought of as an embedded bitstream. The first portion of the bitstream can be used to decode a baseline-quality reconstruction of the original data, without requiring any information from the remainder of the bitstream, and progressively larger portions of the bitstream can be used to decode improved reconstructions of the original data. For purposes of the present Application, progressive encryption is defined as a process which takes original data (plaintext) as

input and creates progressively encrypted data (ciphertext) as output, where the progressively encrypted data has the property that the first portion can be decrypted alone, without requiring information from the remainder of the original data; and progressively larger portions can be decrypted with this same property, in which decryption can require data from earlier but not later portions of the bitstream.

ENCODING METHOD AND SYSTEM

Although specific steps are disclosed in flow chart 600 of Figure 6, such steps are exemplary. That is, the present invention is well suited to performing various other steps or variations of the steps recited in Figure 6. Additionally, for purposes of clarity and brevity, the following discussion and examples will specifically deal with video data. The present invention, however, is not limited solely to use with video data. Instead, the present invention is well suited to use with audio-based data, image-based data, web page-based data, graphic data, and the like. Specifically, the present invention is directed towards any data in which scalable coding is combined with progressive encryption. In step 602 of Figure 6, in one embodiment, the present invention recites receiving video data. In one embodiment, the video data is comprised of a stream of uncompressed video frames which are received by segmenter 702 of the encoder system 700 of Figure 7.

In another embodiment of the present invention, the video data is comprised of prediction error video data generated by a video prediction unit (VPU). As shown Figure 8, in one embodiment of the present invention encoder system 700 has a VPU 800 coupled thereto. VPU 800 generates and forwards prediction error video data to segmenter 702 of encoder system 700. Although VPU 800 of Figure 8 is disposed outside of encoding system 700, the present invention is also well suited to having VPU 800 integral with encoding system 700. Figure 9 illustrates one embodiment of the present invention in which VPU 800 is integral with encoding system 700.

With reference now to step 604 of Figure 6, the present embodiment then segments the received video data into corresponding regions. Figure 10A provides a schematic depiction of a video frame 1000. Video data corresponding to video frame 1000 is received by segmenter 702 of Figures 7, 8, and 9. Figure 10B depicts the same video frame 1000 after segmenter

702 has segmented video frame 1000 into corresponding regions 1002, 1004, 1006, 1008, 1010, and 1012. Although such a quantity and configuration of regions is shown in Figure 10B, such a tiling quantity and configuration is intended to be exemplary only. As one example, Figure 10C illustrates another example of segmentation in which segmenter 702 has segmented video frame 100 into various non-rectangular regions 1014, 1016, 1018, 1020, and 1022. As another example, Figure 10D illustrates another example of segmentation in which segmenter 702 has segmented video frame 100 into various non-rectangular and overlapping regions 1024, 1026, 1028, 1030, and 1032. The overlapping portions are denoted by dotted lines. The present invention is also well suited to an approach in which segmenter 702 has various rectangular regions configured in an overlapping arrangement. Furthermore, the present invention is also well suited to an embodiment in which the regions change from frame to frame. Such an embodiment is employed, for example, to track a foreground person as they move.

Referring now to step 606, encoder 704 of Figures 7, 8 and 9 then scalably encodes the regions into scalable video data. For purposes of the present Application, scalable coding is defined as a process which takes original data as input and creates scalably coded data as output, where the scalably coded data has the property that portions of it can be used to reconstruct the original data with various quality levels. Specifically, the scalably coded data is often thought of as an embedded bitstream. The first portion of the bitstream can be used to decode a baseline-quality reconstruction of the original data, without requiring any information from the remainder of the bitstream, and progressively larger portions of the bitstream can be used to decode improved reconstructions of the original data. That is, separate regions or regions of a video frame are encoded into one or more data packets. The scalable video data generated by the present embodiment has the property that a first small portion of the data can be decoded into baseline quality video, and larger portions can be decoded into improved quality video. It is this property that allows data packets to be transcoded to lower bitrates or spatial resolutions simply by truncating the data packet. This process of truncation will be discussed in further detail below.

With reference still to step 606, in one embodiment of the present invention each region is coded by encoder 704 into two portions: header

data and scalable video data. Hence, in such an embodiment, each data packet contains header data and scalable video data. The header data describes, for example, the region (e.g. the location of the region within the video frame) that the data packet represents and other information used for subsequent transcoding and decoding operations in accordance with the present invention. Furthermore, in one embodiment, the header data contains information including a series of recommended truncation points for data packet transcoders. The scalable video data contains the actual coded video. In the case of intraframe coding, the video data may be the coded pixels; while in the case of interframe coding, it may be the motion vectors and coded residuals that result from motion-compensated prediction. In the present embodiments, scalable coding techniques are used in both cases to create an embedded or scalable data packet that can be truncated to lower the resolution or fidelity of the coded video data. In still another embodiment of the present invention, the scalably encoded video data is prepared by encoder 704 without corresponding header data.

As recited in step 608, the present embodiment then progressively encrypts the scalable video data to generate progressively encrypted scalable video data. That is, packetizer and encrypter 706 of Figures 7, 8, and 9 employs progressive encryption techniques to encrypt the scalable video data. For purposes of the present Application, progressive encryption is defined as a process which takes original data (plaintext) as input and creates progressively encrypted data (ciphertext) as output, where the progressively encrypted data has the property that the first portion can be decrypted alone, without requiring information from the remainder of the original data; and progressively larger portions can be decrypted with this same property, in which decryption can require data from earlier but not later portions of the bitstream. Progressive encryption techniques include, for example, cipher block chains or stream ciphers. These progressive encryption methods have the property that the first portion of the data is encrypted independently, then later portions are encrypted based on earlier portions. When properly matched with scalable coding and packetization, progressive encryption preserves the ability to transcode data packets with simple data packet truncation. More specifically, progressive encryption methods have the property that smaller blocks of data are encrypted progressively. While block code encryption with small block sizes is not very secure, progressive encryption methods add a degree of security by feeding encrypted data of

earlier blocks into the encryption of a later block. Decryption can then be performed progressively as well. In one embodiment, the first small block of ciphertext is decrypted into plaintext by itself while later blocks of ciphertext depend on the decrypted plaintext from earlier blocks. Thus, earlier blocks of ciphertext can be decrypted without knowledge of the entire ciphertext segment. This progressive nature of cipher block chains and stream ciphers matches nicely with the progressive or embedded nature of scalable coding. Although encoding system 700 depicts a combined packetizer and encrypter module 706. Such a depiction is exemplary only, as encoding system 700 of the present invention is well suited to having separate and distinct packetizer and encrypter modules.

As was the case in prior art approaches, entire data packets were encrypted with one long block code. As a result, decryption was not possible unless it the data packet was received in its entirety. However, the present invention is using scalable data packets and it is desired to transcode the stream of scalable data packets by data packet truncation. Therefore, the present invention encrypts the data packets in a similarly progressive manner. Hence, unlike conventional approaches, the present invention is data packet loss resilient. That is, should a data packet be lost, decryption of the remaining data packets is not further complicated and is still readily achievable. This combination of scalable encoding and progressive encryption enables the advantageous transcoding operations described in detail below.

With reference still to step 608, in one embodiment of the present invention, while the payload data (i.e. the scalable video data) is encrypted progressively, the header data is left unencrypted so that transcoding nodes can use this information to make transcoding decisions. For example, in one embodiment, the unencrypted header contains information such as recommended truncation points within the encrypted payload data. In another embodiment, this header data is used to achieve near rate distortion (RD)-optimal bitrate reduction by intermediate transcoding nodes. Moreover, in the present embodiment, the transcoding nodes can use the header data to make transcoding decisions without requiring decryption of the progressively encrypted scalable video data or the header data. In yet another embodiment of the present invention the header data is encrypted to add additional security.

Referring now to step 610, the present invention then packetizes the progressively encrypted scalable video data. In one embodiment, a packetizer and encrypter 706 of Figures 7, 8, and 9 combine and packetize the unencrypted header data with the progressively encrypted scalable video data. The resulting secure scalable data packets are then available to be streamed to desired receivers. In another embodiment, packetizer and encrypter 706 packetizes the progressively encrypted scalable video data and the encrypted header data. Furthermore, in an embodiment which does not include header data, packetizer and encrypter 706 packetizes only the progressively encrypted scalable video data.

Encoding system 700 securely and scalably encodes video data. More specifically, encoding system 700 combines scalable coding with progressive encryption techniques. The resulting scalably encoded, progressively encrypted, and packetized video streams have the feature that subsequent transcoding operations such as bitrate reduction and spatial downsampling can be performed (via e.g. data packet truncation or data packet elimination) without decrypting the packetized data and thus while maintaining the security of the system. The present invention is also well suited to an embodiment in which only some, but not all, of the regions formed by segmenter 702 are ultimately forwarded from encoding system 700. As an example, in one embodiment of the foreground of a video data image is forwarded, as the background image may not have changed since a previous transmission, or perhaps the background image does not contain data of interest.

DECODING METHOD AND SYSTEM

Although specific steps are disclosed in flow chart 1100 of Figure 11, such steps are exemplary. That is, the present invention is well suited to performing various other steps or variations of the steps recited in Figure 11. In step 1102 of Figure 11, the present invention receives a data packet containing progressively encrypted and scalably encoded video data. More specifically, decrypter 1202 of decoding system 1200, both of Figure 12, receives the data packet containing progressively encrypted and scalably encoded video data. In one embodiment, the received data packet also includes header data wherein the header data provides information corresponding to the scalably encoded video data. In yet another embodiment, the received data packet also includes encrypted header data providing information corresponding to the scalably encoded video data.

As recited in step 1104, the present invention then decrypts the data packet containing the progressively encrypted and scalably encoded video data to generate scalably encoded regions. That is, decrypter 1202 of Figure 12 decrypts the progressively encrypted and scalably encoded video data to generate scalably encoded regions. Furthermore, in an embodiment in which the received data packet includes encrypted header data, decrypter 1202 also decrypts the encrypted header data.

Referring now to step 1106, the present embodiment then decodes the scalably encoded regions to provide decoded regions. As described above in conjunction with the description of encoding system 700 of Figures 7, 8, and 9, a video frame 1000 as shown in Figure 10A can be segmented in multiple corresponding regions 1002, 1004, 1006, 1008, 1010, and 1012 as shown in Figure 10B.

At step 1108, the present invention then assembles the decoded regions to provide video data. Moreover, assembler 1206 of decoding system 1200 of Figure 12 assembles the decoded regions to provide video data. In one embodiment of the present invention decoding system 1200 then provides as output, video data in the form of an uncompressed video stream. In another embodiment of the present invention, assembler 1206 outputs video data comprised of prediction error video data suitable for by a video prediction unit (VPU). As shown Figure 13, in one embodiment of the present invention decoder system 1200 has a VPU 1300 coupled thereto. VPU 1300 uses the output of assembler 1206 to ultimately provide an uncompressed stream of video frame data. Although VPU 1300 of Figure 13 is disposed outside of decoding system 1200, the present invention is also well suited to having VPU 1300 integral with decoding system 1200. Figure 14 illustrates one embodiment of the present invention in which VPU 1300 is integral with decoding system 1200. Hence, the present invention provides a method and system for decoding video data which has been securely and scalably encoded.

TRANSCODING METHOD AND SYSTEM

Figure 15A is a block diagram of an exemplary hybrid wired/wireless network 1500 upon which embodiments of the present invention may be practiced. In hybrid wired/wireless network 1500, media (e.g., video) data are streamed to fixed clients (stationary receiving

nodes) via a wired link and to mobile clients (moving receiving nodes) via a wireless link.

In the present embodiment, hybrid wired/wireless network 1500 includes a wired sender (source 1510), a wired high-resolution receiver 1520, and a wireless medium-resolution receiver 1540. In this system, source 1510 generates a full-bandwidth, high-resolution video stream 1550a that is sent to high-resolution receiver 1520. A transcoder 1530, placed at source 1510, at medium-resolution receiver 1540, or at an intermediate node such as a wired/wireless gateway, transcodes the stream 1550a into a lower-bandwidth, medium-resolution video stream 1550b which is then sent to medium-resolution receiver 1540.

Figure 15B is a block diagram of an exemplary wireless network 1501 (e.g., a wireless appliance network) upon which embodiments of the present invention may be practiced. In wireless appliance networks, mobile senders and receivers communicate with one another over wireless links. A sender's coverage area is limited by the power of the transmitted signal. Relay devices can be used to extend the wireless coverage area when intended receivers are beyond the immediate coverage area of the sender. In the case of heterogeneous receivers (e.g., receiving nodes having different display, power, computational, and communication characteristics and capabilities), transcoders can be used to adapt a video stream for a particular receiver or communication link. Transcoding can be performed in a relay device or in a receiver which also acts as a relay. Transcoding can also be performed by the sender or by the receiving node.

In the present embodiment, wireless network 1501 includes a wireless sender (source 1510), a high-resolution receiver and transcoder 1560, and a medium-resolution (lower bandwidth) receiver 1540. In wireless network 1501, the high-resolution receiver 1560 receives and transcodes the high-resolution video stream 1550a, and relays the resulting lower-bandwidth stream 1550b to the medium-resolution receiver 1540.

Referring to Figures 15A and 15B, both hybrid wired/wireless network 1500 and wireless network 1501 use network transcoders to transcode video streams 1550a into lower bandwidth streams 1550b that

match the display capabilities of the target wireless nodes (e.g., medium-resolution receiver 1540). Generally speaking, these networks illustrate how network transcoding can enable efficient use of wireless spectrum and receiver resources by transcoding media (e.g., video) streams into formats better suited for transmission over particular channels and for the capabilities of the receiving nodes.

Figure 16 is a block diagram of a system 1600 including a source node 1610, an intermediate (transcoder) node 1620, and a receiving node 1630 in accordance with one embodiment of the present invention. In this embodiment, transcoder 1620 is a separate node transposed between source node 1610 and receiving node 1630. However, the functions performed by transcoder 1620 may instead be performed by source node 1610 or by receiving node 1630.

In the present embodiment, source node 1610 encodes and/or encrypts a stream of data packets and sends these data packets to transcoder 1620, as described above. In one embodiment, each of the data packets in the stream has a header portion and a payload portion (see Figure 20, below); in another embodiment, the data packet has only a payload portion (see Figure 21, below). The payload portion carries the media data (e.g., video data), while the header portion carries information that is used by transcoder 1620 to transcode the payload portion. A data packet, including the information carried by the header portion, and the transcoding method used by transcoder 1620 are further described below. In one embodiment, only the payload portion is encrypted and encoded. In another embodiment, the payload portion is encrypted and encoded, and the header portion is also encrypted.

In the present embodiment, transcoder 1620 performs a transcoding function on the data packets received from source node 1610. The transcoding function performed by transcoder 1620 is described in conjunction with Figure 19, below. The purpose of the transcoding function is to configure the stream of data packets according to the attributes downstream of transcoder 1620, such as the attributes of the receiving node 1630 or the attributes of communication channel 1625 linking transcoder 1620 and receiving node 1630. The transcoding function can include, for example, truncation of the data packets or elimination of certain data packets from the stream. In the case in which

the stream is already configured for the receiving node 1630 or for communication channel 1625, the transcoding function consists of a pass-through of the data packets in the stream without modification.

Of particular significance, in accordance with the present invention, transcoder 1620 performs a transcoding function without decrypting and/or decoding the data packets (specifically, the media data in the data packets). In the embodiment in which the data packets have a header portion and a payload portion, and where the header portion is encrypted, transcoder 1620 only decrypts the header portion. In either case, in comparison to a conventional transcoder, transcoder 1620 of the present invention requires less computational resources because there is no need to decrypt the media data. In addition, the present invention provides end-to-end security while enabling very low complexity transcoding to be performed at intermediate, possibly untrusted, nodes without compromising the security of the media data.

Continuing with reference to Figure 16, transcoder 1620 has knowledge of the attributes of receiving node 1630 and/or communication channel 1625. These attributes include, but are not limited to, the display, power, communication and computational capabilities and characteristics of receiving node 1630, or the available bandwidth on communication channel 1625. For example, in one embodiment, transcoder 1620 receives the attribute information from receiving node 1630, or transcoder 1620 reads this information from receiving node 1630. In another embodiment, transcoder 1620 may be implemented as a router in a network; the router can determine if there is congestion on the next "hop" and transcode the stream of data packets accordingly.

In the present embodiment, after transcoding, transcoder 1620 sends the resultant stream of data packets, comprising the encoded and encrypted media data packets, to receiving node 1630.

Figure 17 is a block diagram of one embodiment of a transcoder device 1620 upon which embodiments of the present invention may be practiced. In this embodiment, transcoder 1620 includes a receiver 1710 and a transmitter 1720 for receiving a stream of data packets from source node 1610 (Figure 16) and for sending a stream of data packets to receiving node 1630 (Figure 16), respectively. Receiver 1710 and transmitter 1720 are

capable of either wired or wireless communication. Separate receivers and transmitters, one for wired communication and one for wireless communication, may also be used. It is appreciated that receiver 1710 and transmitter 1720 may be integrated as a single device (e.g., a transceiver).

Continuing with reference to Figure 17, transcoder device 1620 may include an optional controller 1730 (e.g., a processor or microprocessor), an optional decrypter 1740, and an optional memory 1750, or a combination thereof. In one embodiment, decrypter 1740 is used to decrypt header information. In another embodiment, memory 1750 is used to accumulate data packets received from source node 1610 before they are forwarded to receiving node 1630 (Figure 16).

Figures 18A, 18B, 18C, 18D and 18E are data flow diagrams illustrating various embodiments of a method for transcoding data packets in accordance with the present invention. In the embodiments of Figures 18A-D, the data packets each have a header portion and a payload portion; in the embodiment of Figure 18E, the data packets do not have a header portion. In each of the embodiments of Figures 18A-E, the data packets (specifically, the media data) are encrypted and may be encoded. The embodiments of Figures 18A-E are separately described in order to more clearly describe certain aspects of the present invention; however, it is appreciated that the present invention may be implemented by combining elements of these embodiments.

In accordance with the present invention, the method for transcoding data packets is performed on the encrypted data packets; that is, the media data are not decrypted. Transcoding functions can include truncation of the data packets (specifically, the payload portions of the data packets), eliminating certain data packets from the stream, or passing the data packets through without modification.

With reference first to Figure 18A, incoming encrypted and/or encoded data packets are received by transcoder 1620. In this embodiment, the header portion of each data packet is not encrypted. Transcoder 1620 reads the header portion, which contains information that can be used to make transcoding decisions. In one embodiment, the information in the header portion includes specification of the truncation

points. In another embodiment, the truncation points are derived from the information provided in the header.

For example, the header portion may contain information specifying recommended points (e.g., a number of a bit) for truncating the payload portion of the data packets. It is appreciated that each data packet may have a different truncation point. The recommended truncation point can be selected using a variety of techniques. In one embodiment, the truncation point for each data packet is specified according to an analysis such as a rate-distortion (RD) analysis, so that the stream of data packets can be compressed to a rate that is RD optimal or near-RD optimal. In another embodiment, the header portion contains information that describes the RD curves generated by the RD analysis, and the truncation points are derived from further analysis of the RD curves.

In the present embodiment, RD optimal coding is achieved by generating an RD plot for each region of a video image, and then operating on all regions at the same slope that generates the desired total bitrate. Near-optimal transcoding can be achieved at the data packet level by placing the optimal RD cutoff points for a number of quality levels in the header portions of the data packets. Then, transcoder 1620 (Figure 16) can truncate each packet at the appropriate cutoff point; thus, the resulting packets will contain the appropriate number of bits for each region of the image for the desired quality level. Transcoder 1620 reads each packet header, then truncates the packet at the appropriate point. For example, if three regions in an image are coded into separate packets, for each region three RD optimal truncation points are identified and their locations placed in the respective packet header. Transcoder 1620 can choose to operate at any of the three RD points (or points in between), and then can truncate each packet at the appropriate cutoff point.

The header portion may also contain information identifying each data packet by number, for example. Accordingly, transcoder 1620 can eliminate certain data packets from the stream; for example, if every other packet is to be eliminated (e.g., the odd-numbered packets), transcoder 1620 can use the header information to identify the odd-numbered data packets and eliminate those from the stream of data packets.

The embodiment of Figure 18B is similar to that of Figure 18A, except that the header portion of each data packet is encrypted. In this case, transcoder 1620 first decrypts the header portion, before reading the header information and operating on the stream of data packets as described above.

In the embodiment of Figure 18C, data packets are accumulated in memory. That is, instead of a first-in/first-out type of approach, a subset of the data packets in the stream is accumulated and stored in memory (e.g., memory 1750 of Figure 17) before they are forwarded to the receiving node. In this embodiment, the header information for all of the accumulated data packets in the subset is used to make transcoding decisions. The transcoding decisions are made based on the attributes of the receiving node 1630 or the attributes of the communication channel 1625 (Figure 16), as described previously herein. It may be possible, and perhaps desirable, to configure the stream of data packets according to the attributes of the receiving node or communication channel without operating on every data packet in the stream. For example, instead of truncating all of the data packets in the subset, a decision may be made to truncate only a portion of the packets in the subset, or to truncate the packets at a point other than the recommended truncation point.

In the embodiment of Figure 18D, transcoder 1620 receives information from the downstream receiving node (e.g., receiving node 1630 of Figure 16). In one embodiment, the information describes attributes of receiving node 1630, such as its display, power, computational and communication capabilities and characteristics. Based on the information received from receiving node 1630, transcoder 1620 can make transcoding decisions based on the information in the header portions of the data packets. For example, transcoder 1620 can pick a truncation point depending on whether receiving node 1630 is a medium- or low-resolution device, and transcoder 1620 can choose not to modify the stream of data packets if receiving node 1630 is a high-resolution device. Similarly, transcoder 1620 can receive information describing the attributes of communication channel 1625 (Figure 16)

In the embodiment of Figure 18E, the incoming data packets do not have a header portion. Accordingly, transcoder 1620 makes transcoding

decisions based on a pre-defined set of rules. That is, instead of truncating each data packet at a different point specified by the information in the header portion, transcoder 1620 may truncate all data packets in the stream at the same point, depending on the attributes of the receiving node or communication channel.

Figure 19 is a flowchart of the steps in a process 1900 for transcoding data packets in accordance with one embodiment of the present invention. In one embodiment, process 1900 is implemented by transcoder device 1620 (Figure 17) as computer-readable program instructions stored in memory 1750 and executed by controller 1730. Although specific steps are disclosed in of Figure 19, such steps are exemplary. That is, the present invention is well suited to performing various other steps or variations of the steps recited in Figure 19.

In step 1910 of Figure 19, a stream of data packets is received from a source node (e.g., source 1610 of Figure 16). In the present embodiment, the data packets include encrypted media data (e.g., video data). In one embodiment, the media data are also encoded. In another embodiment, the data packets include a header portion and a payload portion. In one embodiment, the header portion is also encrypted.

In step 1915 of Figure 19, in one embodiment, information describing the attributes of a downstream receiving node (e.g., receiving node 1630 of Figure 16) or communication channel (e.g., communication channel 1625 of Figure 16) is received. In another embodiment, the attributes of receiving node 1630 or communication channel 1625 are already known.

In step 1920 of Figure 19, a transcoding function is performed on the stream of data packets to configure the stream according to the attributes of receiving node 1630. Significantly, the transcoding function is performed without decrypting the media data in the data packets. In one embodiment, the transcoding function is performed on information provided by the header portion of each data packet. In one such embodiment, the header information provides recommended truncation points for the payload portion of the respective data packet. In another embodiment, the truncation points are derived from the information provided in the header portion.

In step 1922, in one embodiment, the transcoding function eliminates certain data packets from the stream. In step 1924, in one embodiment, the transcoding function truncates the media data in the data packets. It is appreciated that each data packet may have a different truncation point. In step 1926, in one embodiment, the transcoding function passes the data packets through without modification.

In step 1930, the transcoded data packets (still encrypted and/or encoded) are sent to receiving node 1630.

In summary, the above-listed embodiment of the present invention provides a secure method and system for transcoding media data for a variety of downstream attributes, such as the attributes of receiving nodes having different capabilities and characteristics or the attributes of the communication between the transcoder and a receiving node. Because the encrypted media data do not need to be decrypted and then encrypted again, the computational resources needed for transcoding the stream of data packets is significantly reduced, and the security of the media data is not compromised.

SECURE SCALABLE DATA PACKET

With reference now to Figure 20, a schematic representation of a data packet 2000 formed in accordance with one embodiment of the present invention is shown. Furthermore, as mentioned above, for purposes of clarity and brevity, the following discussion and examples will specifically deal with video data. The present invention, however, is not limited solely to use with video data. Instead, the present invention is well suited to use with audio-based data, image-based data, web page-based data, and the like. It will be understood that in the present embodiments, data packet 2000 is generated by encoding system 700 of Figures 7, 8, and 9, operated on by transcoder 1620 of Figures 16, 18A, 18B, 18C, 18D, and 18E , and then ultimately forwarded to decoding system 1200 of Figures 12, 13, and 14. During the aforementioned process, data packet 2000 is stored on computer readable media residing in, and causes a functional change or directs the operation of, the devices (e.g. general purpose networked computer systems, embedded computer systems, routers, switches, server devices, client devices, various intermediate devices/nodes, stand alone computer systems, and the like) in which, for

example, transcoder 1620 and/or decoder 1200 are implemented.

In the embodiment of Figure 20, data packet 2000 includes header data portion 2002 and scalably encoded, progressively encrypted video data portion 2004. As mentioned above, header data portion 2002 includes information that is used by transcoder 1620 to transcode the scalably encoded, progressively encrypted video data portion 2004. For example, header data portion 2002 may contain information specifying recommended points (e.g., a number of a bit) for truncating the payload portion (i.e. the scalably encoded, progressively encrypted video data portion 2004) of data packet 2000. Header data portion 2002 may also contain information identifying each data packet by number, for example. Accordingly, transcoder 1620 can eliminate certain data packets from the stream; for example, if every other packet is to be eliminated (e.g., the odd-numbered packets), transcoder 1620 can use the information in header data portion 2002 to identify the odd-numbered data packets and eliminate those from the stream of data packets.

With reference still to Figure 20, data packet 2000 also includes potential truncation points 2006, 2008, and 2010 within scalably encoded, progressively encrypted video data portion 2004. Although such truncation points are shown in Figure 20, the configuration of truncation points 2006, 2008, and 2010, is exemplary only. That is, the present invention is well suited to having a lesser of greater number of truncation points, and to having the truncation points located other than where shown in Figure 20. Again, as mentioned above, truncation points 2006, 2008, and 2010 are used by transcoder 1620 during its operation on packet 2000. Additionally, in one embodiment of the present invention, header data portion 2002 is encrypted.

In the embodiment of Figure 21, data packet 2100 does not include a header data portion, and instead includes only scalably encoded, progressively encrypted video data portion 2104. With reference still to Figure 21, data packet 2100 also includes potential truncation points 2104, 2106, and 2108 within scalably encoded, progressively encrypted video data portion 2104. Although such truncation points are shown in Figure 21, the configuration of truncation points 2104, 2106, and 2108, is exemplary only. That is, the present invention is well suited to having a lesser of greater number of truncation points, and to having the truncation points

located other than where shown in Figure 21. Again, as mentioned above, truncation points 2104, 2106, and 2108 are used by transcoder 1620 during its operation on packet 2100.

Thus, the present invention provides, in one embodiment, a data packet format which enables secure and scalable encoding, transcoding, and decoding of data.

The foregoing descriptions of specific embodiments of the present invention have been presented for purposes of illustration and description. They are not intended to be exhaustive or to limit the invention to the precise forms disclosed, and obviously many modifications and variations are possible in light of the above teaching. The embodiments were chosen and described in order to best explain the principles of the invention and its practical application, to thereby enable others skilled in the art to best utilize the invention and various embodiments with various modifications as are suited to the particular use contemplated. It is intended that the scope of the invention be defined by the Claims appended hereto and their equivalents.